Fondamenti di informatica 1 (informatici) — A.A. 1999-2000
Appello del 19 settembre 2000
Tempo a disposizione: 60 minuti

SommeAlterne — Analisi di un array di array

Cognome:_____ Nome: _____ Matricola:_____ Calcolatore: _____

Specifiche:

Sia $\mathbf{x_0}$, $\mathbf{x_1}$, $\mathbf{x_2}$, **...** una sequenza numerica; la **somma alterna** di tale sequenza è $\mathbf{x_0 - x_1 + x_2 - x_3 + ...}$, che è data dalla "somma" degli elementi della sequenza in cui gli elementi vengono alternativamente sommati (quelli di indice pari) e sottratti (quelli di indice dispari). Ad esempio, la somma alterna della sequenza **1**, **–2**, **4**, **5**, **7**, **8** vale **1** (**1–(–2)+4–5+7–8**).

Nell'ambito della classe Java **SommeAlterne**, definire i seguenti metodi di classe:

– un metodo **sommaAlterna** che, ricevendo come parametro un array **a** (non nullo) di interi, calcola e restituisce la somma alterna degli elementi di **a**;

– un metodo **sommeAlterneCostanti** che, ricevendo come parametro un array **m** (non nullo) di array (non nulli) di interi, verifica se tutte le righe di **m** hanno la stessa somma alterna, e restituisce l'esito di tale verifica.

Esempi:

Si considerino gli array **a** = **{ 1, –2, 3 }**, **b** = **{ 2, 2 }** e **c** = **{ }**

– l'invocazione **SommeAlterne.sommaAlterna(a)** deve restituire **6**

– le invocazioni **SommeAlterne.sommaAlterna(b)** e **SommeAlterne.sommaAlterna(c)** devono restituire **0**

Si considerino gli array di array **m1** = **{ { 2, 1 }, { 1, 1, 1 }, { 1 } }** e **m2** = **{ { 2, 1 }, { 0, –1 }, { } }**

– l'invocazione **SommeAlterne.sommeAlterneCostanti(m1)** deve restituire **true** — infatti, la somma alterna di ciascuna delle tre righe di **m1** vale **1**

– l'invocazione **SommeAlterne.sommeAlterneCostanti(m2)** deve restituire **false** — infatti, le prime due righe di **m2** hanno somma alterna **1**, ma la terza riga ha somma alterna **0**, che è quindi diversa da **3**

Che cosa deve essere fatto:

Sulla base delle specifiche dell'esercizio, definire la classe **SommeAlterne** come segue:

– definire il metodo di classe **int sommaAlterna(int[] a)** — si assuma **a** non nullo;

– definire il metodo di classe **boolean sommeAlterneCostanti(int[][] m)** — si assuma **m** non nullo, e ogni riga di **m** non nulla;

– definire una applicazione di test

– definire, se necessario, altri metodi privati.

Verifica:

La correttezza del codice scritto deve essere verificata mediante compilazione ed esecuzione, utilizzando degli opportuni insiemi di dati di ingresso.

Ciascuna degli insiemi di dati di ingresso usato per la verifica deve essere documentato da una invocazione di metodo e da un commento nel codice, che descrive l'insieme di dati di ingresso e motiva la scelta dell'insieme di dati di ingresso. Ad esempio:

```
/* calcola la somma alterna di un array non vuoto */
System.out.println(somma(new int[] {1, -2, 3}));   // 6
/* calcola sommeAlterneCostanti nel caso in cui una riga sia vuota
 * e tutte le righe abbiano somma alterna 0 */
System.out.println(
    sommeAlterneCostanti(new int[][] {{0}, {}, {1, 1}}));   // true
```

Fondamenti di informatica 1 (informatici) — A.A. 1999-2000
Appello del 19 settembre 2000
Tempo a disposizione: 60 minuti

Frequenze — Analisi di un array di array

Cognome:_____ Nome: _____ Matricola:_____ Calcolatore: _____

Specifiche:

Nell'ambito della classe Java **Frequenze**, definire i seguenti metodi di classe:

- un metodo **frequenza** che, ricevendo come parametri un array **a** (non nullo) di interi e un intero **k**, calcola e restituisce la frequenza di **k** in **a**, ovvero il numero degli elementi di **a** uguali a **k**;
- un metodo **frequenzeCostanti** che, ricevendo come parametri un array **m** (non nullo) di array (non nulli) di interi e un intero **k**, verifica se **k** occorre in tutte le righe di **m** con la stessa frequenza, e restituisce l'esito di tale verifica.

Esempi:

Si considerino gli array **a** = **{ 1, 2, 1, 3 }** e **b** = **{ }**

- l'invocazione **Frequenze.frequenza(a, 1)** deve restituire **2**, **Frequenze.frequenza(a, 2)** deve restituire **1**, e **Frequenze.frequenza(a, 8)** deve restituire **0**
- l'invocazione **Frequenze.frequenza(b, 1)** deve restituire **0**

Si considerino gli array di array **m1** = **{ { 1, 2 }, { 1, 2, –1 }, { 2 } }** e **m2** = **{ { 1, 2 }, { 0, 3 }, { } }**

- l'invocazione **Frequenza.frequenzeCostanti(m1, 2)** deve restituire **true** — infatti, il valore **2** occorre con frequenza **1** in ciascuna delle tre righe di **m1**
- l'invocazione **Frequenza.frequenzeCostanti(m1, 1)** deve restituire **false** — infatti, il valore **1** occorre con frequenza **1** nella prima e seconda riga di **m1**, ma con frequenza **0** nella terza riga
- l'invocazione **Frequenza.frequenzeCostanti(m2, 8)** deve restituire **true** — infatti, il valore **8** occorre con frequenza **0** in ciascuna delle tre righe di **m2**

Che cosa deve essere fatto:

Sulla base delle specifiche dell'esercizio, definire la classe **Frequenze** come segue:

- definire il metodo di classe **int frequenza(int[] a, int k)** — si assuma **a** non nullo;
- definire il metodo di classe **boolean frequenzeCostanti(int[][] m, int k)** — si assuma **m** non nullo, e ogni riga di **m** non nulla;
- definire una applicazione di test
- definire, se necessario, altri metodi privati.

Verifica:

La correttezza del codice scritto deve essere verificata mediante compilazione ed esecuzione, utilizzando degli opportuni insiemi di dati di ingresso.

Ciascuna degli insiemi di dati di ingresso usato per la verifica deve essere documentato da una invocazione di metodo e da un commento nel codice, che descrive l'insieme di dati di ingresso e motiva la scelta dell'insieme di dati di ingresso. Ad esempio:

```
/* calcola la frequenza nel caso in cui k occorre solo come ultimo elemento */
System.out.println(frequenza(new int[] {1, 2, 3}, 3));   // 1
/* calcola frequenzeCostanti nel caso in cui una riga sia vuota
 * e l'elemento non occorra in nessuna riga */
System.out.println(
    frequenzeCostanti(new int[][] {{0}, {}, {1, -1} }, 8)); // true
```